INFO 6210 Data Management and Database Design

AI Skunkworks Project | Hyperparameter Database|DB14

Megha Ponneti Nanda|ponnetinanda.m@husky.neu.edu
Pratiksha Milind Lavhatre|lavhatre.p@husky.neu.edu

Abstract

Gather a list of data sets, type of datasets, and hyperparameters by running an expanded list of datasets. This information will be embedded in a database management system, to be incorporated into a website where it is easy to be searched and used by the public.

The hyperparameter database is created by running millions of hyperparameter values, over thousands of public datasets and calculating the individual conditional expectation of every hyperparameter on the quality of a model.

Generate models using H2O software to find the best hyperparameters and create a conceptual model and store all the data into a physical database.

Introduction

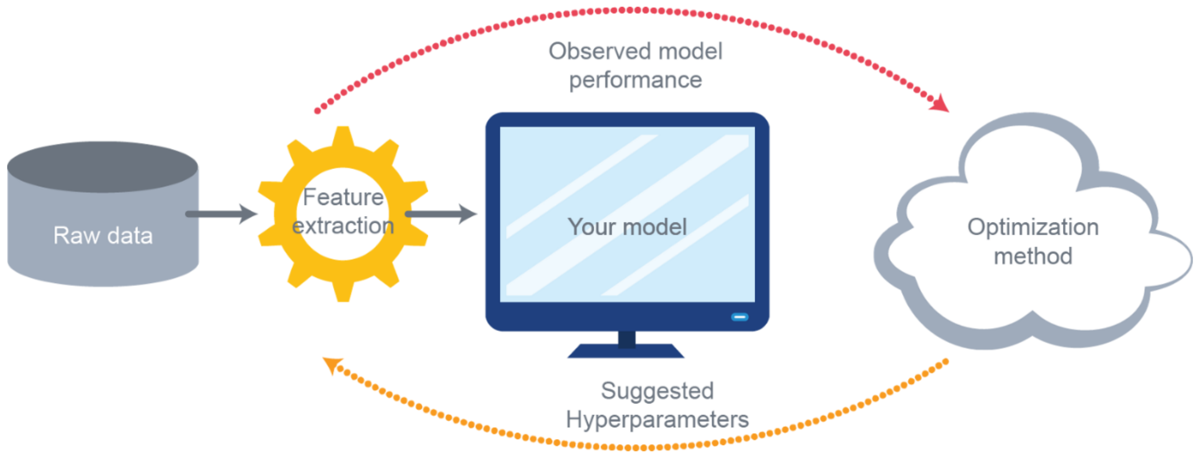
Hyperparameters are parameters that are specified prior to running machine learning algorithms that have a large effect on the predictive power of statistical models. Knowledge of the relative importance of a hyperparameter to an algorithm and its range of values is crucial to hyperparameter tuning and creating effective models.

The hyperparameter database is a public resource with algorithms, tools, and data that allows users to visualize and understand how to choose hyperparameters that maximize the predictive power of their models.

The hyperparameter database is created by running millions of hyperparameter values, over thousands of public datasets and calculating the individual conditional expectation of every hyperparameter on the quality of a model.

Currently, the hyperparameter database analyzes the effect of hyperparameters on the following algorithms: Distributed Random Forest (DRF), Generalized Linear Model (GLM), Gradient Boosting Machine (GBM). Naïve Bayes Classifier, Stacked Ensembles, Xgboost and Deep Learning Models (Neural Networks).

The hyperparameter database also uses these data to build models that can predict hyperparameters without search and for visualizing and teaching statistical concepts such as power and bias/variance tradeoff.



Objective

To create Hyperparameter Database by running several hyperparameter values on several datasets and to calculate the individual conditional expectations on every hyperparameter on quality of model

Project Requirement:

Unique datasets are to be picked from different data sources like Kaggle Datasets, UCI machine learning repository, Amazon Datasets, Google Datasets, Computer Vision Datasets etc. Identify the type of dataset chosen, ie Regression, Classification, Clustering etc. Perform data cleaning and data pre-processing. Create conceptual and ER diagrams. Perform database normalization and perform analytics on the database created to get the best values for the hyperparameters.

Problems to be addressed:

Most of the algorithms that improve metrics, degrades the quality of search results. Hyperparameter optimization is performed to overcome the issues addressed by those algorithms and build models for visualizing and teaching statistical concepts.

Potential pitfalls & challenge:

Different optimization methods will have different setup steps, time requirements, and performance outcomes. Hence, methods like algorithmic optimization will help in achieving better performance.

Data source

Data is derived from Kaggle Datasets. We have chosen considered dataset of Travel Insurance.

https://www.kaggle.com/mhdzahier/travel-insurance

A third-party TRAVEL INSURANCE servicing company that is based in Singapore.

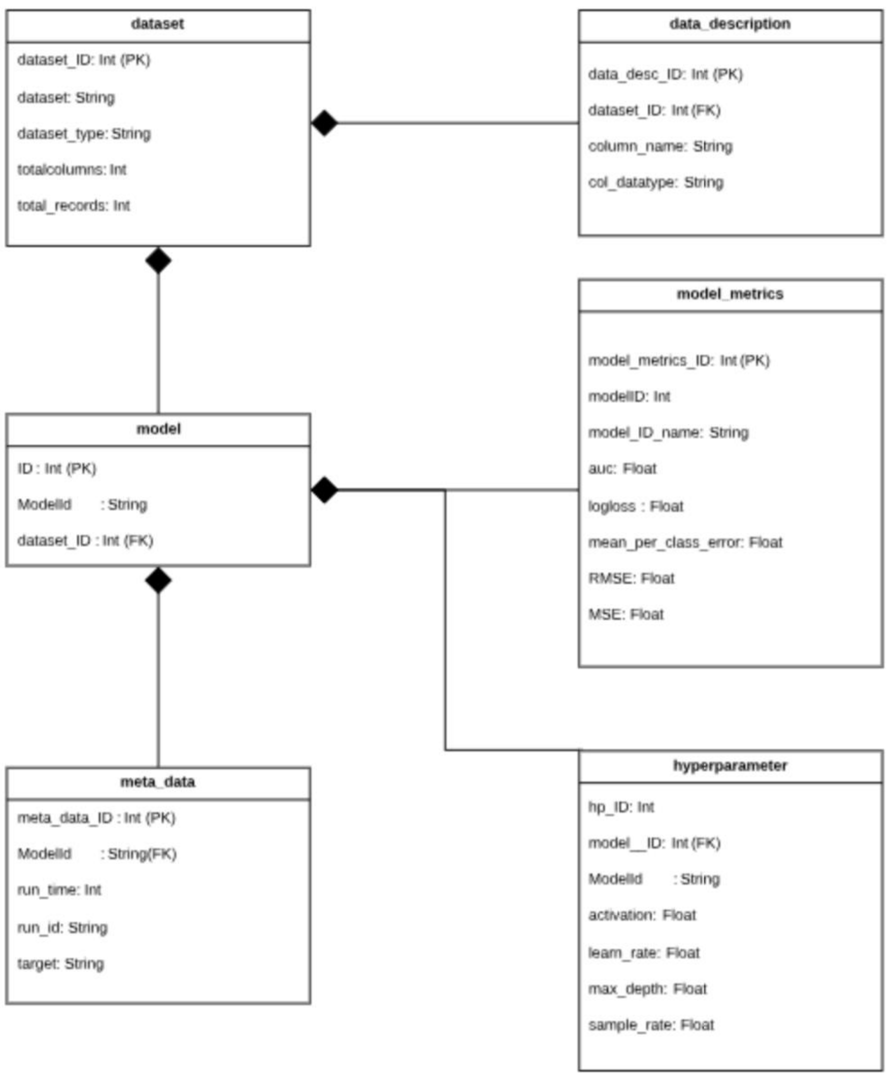
The attributes:

- Target: Claim Status
- Name of agency
- Type of travel insurance agencies
- Distribution channel of travel insurance agencies
- Name of the travel insurance products
- Duration of travel
- Destination of travel
- Amount of sales of travel insurance policies
- Commission received for travel insurance agency
- Gender of insured
- Age of insured

Conceptual Schema

A conceptual data model identifies the highest-level relationships between the different entities.

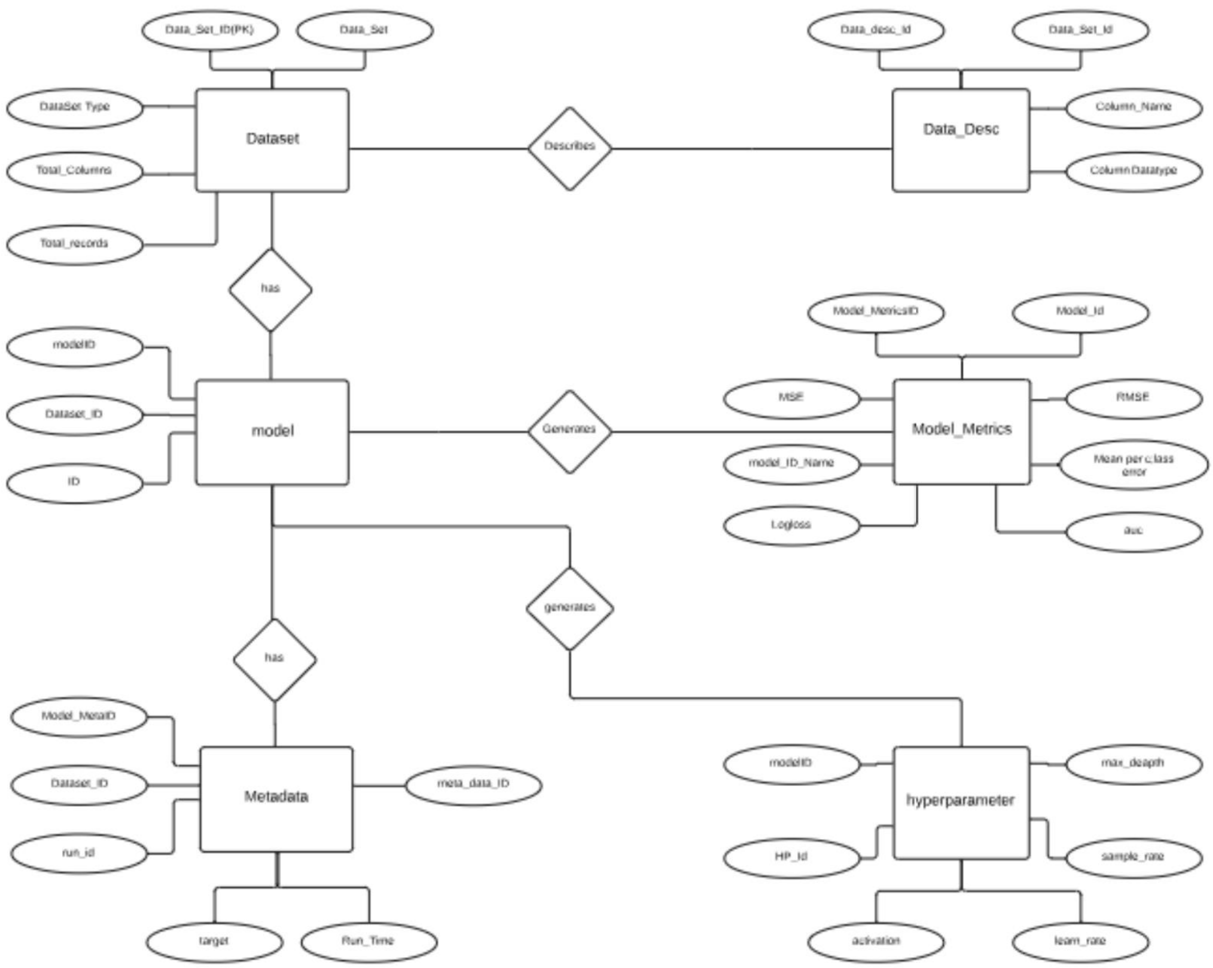
- dataset_ID is the primary key of dataset table.
- data_desc_ID is the primary key and dataset_ID is the foreign key of data_description table.
- meta_data_ID is the primary key and dataset_ID is the foreign key of meta_data table.
- model_metrics_ID is the primary key and meta_data_ID is the foreign key of model_metrics table.
- hp_ID is the primary key and model_metrics_ID is the foreign key of hyperparameter table.



ER Diagram

An entity relationship diagram shows the relationships of entity sets stored in a database. An entity is an object, a component of data. An entity set is a collection of similar entities. These entities have attributes that define its properties.

Entity dataset describes the entity data_description and has the entity meta_data. Entity meta_data generates entity model_metrics. Entity model_metrics generates hyperparameter entity.



Normalization

According to 1 Normal Form (1NF),

1. There are no repeating groups
2.Maintained atomic data values of hyperparameter table is further split into hyperparameter_default and hyperparameter_actual columns
3. Each field of the table has unique name
4. Each table has primary key

According to 2 Normal Form (2NF),

1. All tables satisfy 1NF
2. All non-key attributes are dependent on all parts of primary key. Thus, no partial dependencies
3. There are no calculated data
4. Each field of the table has unique name
5. Each table has primary key

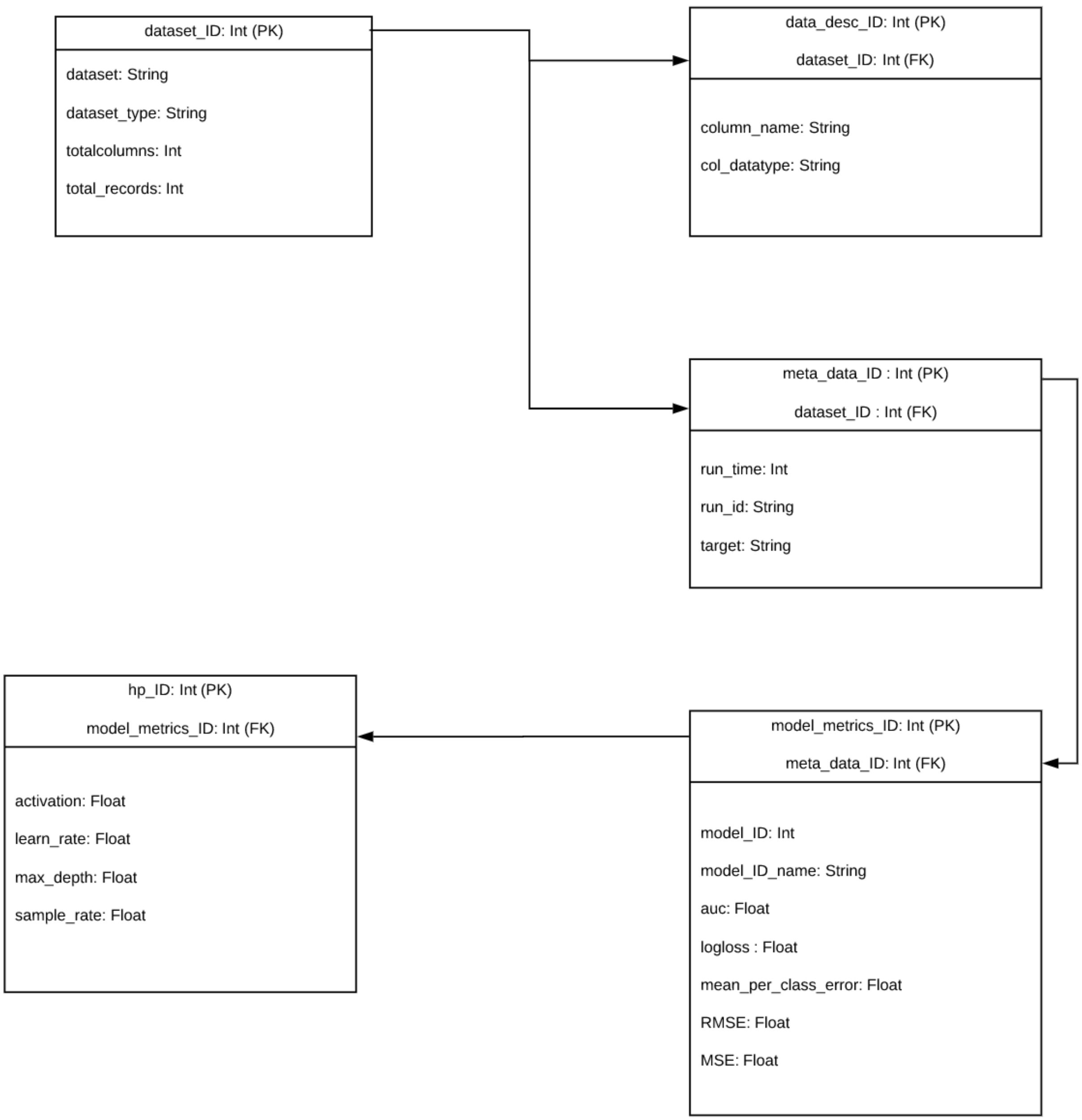
According to 3 Normal Form (3NF),

1. All tables satisfy 2NF
2. All non-key attributes are not dependent on other non-key attributes. Thus, no transitive relationship
3. Each field of the table has unique name
4. Each table has primary key

Physical Model

Physical data model represents how the model will be built in the database. A physical database model shows all table structures, including column name, column data type, column constraints, primary key, foreign key, and relationships between tables.

All the entities from ER diagram is converted to tables. All the attributes are converted into columns of respective tables. All the relationships are converted into foreign keys.



Use Cases

1. Number of 'run time' for dataset 'Travel Insurance'
2. Number of models for run time '1000'
3. Return the actual value of learn_rate for model ID 1
4. What are the metric values of the learn_rate of model 1 of run time 1000
5. Which run time has the highest RMSE value
6. Which run time has the least logloss value
7. Total number of models generated for all the runs
8. Auc value for model 3 of run time 1500
9. Display the number of models for each run times
10. Which Run has the best model
11. For run time 1500 which model ID has the ideal AUC value

Analytics

Below are the queries for the above use cases

```sql
-- Number of 'run time' for dataset 'Travel Insurance'
select count(run_time) from hyperparameter_db14.meta_data;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

count(run_time)
5

```sql
-- Number of models for run time '1000'
select count(model_ID) from hyperparameter_db14.model_metrics m, hyperparameter_db14.meta_data md
where m.meta_data_ID=md.meta_data_ID
and m.meta_data_ID= (select meta_data_ID from hyperparameter_db14.meta_data where run_time=1000);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

count(model_ID)
25

```sql
--      To  return the actual value learn_rate for model id 1

select learn_rate_actual from hyperparameter h, model_metrics mm
where h.model_metrics_ID= mm.model_metrics_ID
and mm.model_id=1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

learn_rate_actual
0.1
0.1
0.01
0.008
0.01

```sql
-- What are the metric values of the learn_rate hyperparameter of model 1 for run time 1000
select RMSE, MSE, mean_per_class_error,logloss,auc from hyperparameter_db14.model_metrics mm,meta_data md,
 hyperparameter_db14.hyperparameter h
where mm.model_metrics_ID=h.model_metrics_ID and
md.meta_data_ID=mm.meta_data_ID and  mm.model_ID=1 and run_time=1000;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

RMSE	MSE	mean_per_class_error	logloss	auc
0.118876642	0.014131656	0.335053974	0.065743315	0.825466329

```sql
-- Which run time has the highest RMSE value
select run_time from hyperparameter_db14.meta_data m, hyperparameter_db14.model_metrics mm
where m.meta_data_ID=mm.meta_data_ID
and RMSE= (select max(RMSE) from hyperparameter_db14.model_metrics);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

run_time
2000

```sql
-- Which run time has the least logloss value
select run_time from hyperparameter_db14.meta_data md, hyperparameter_db14.model_metrics mm
where md.meta_data_ID=mm.meta_data_ID
and logloss= (select min(logloss) from hyperparameter_db14.model_metrics);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

run_time
2500

```sql
-- 10.  Total number of model generated for all the runs
select count(model_ID) from model_metrics
```

count(model_ID)
175

```sql
-- Auc value  for model 3 of run time 1500
select auc from model_metrics mm, meta_data md
 where mm.meta_data_ID=md.meta_data_ID and model_ID=3 and run_time=1500;
```

auc
0.827183564

```sql
--  Display the number of models for eachrun times
 select count(model_ID),mm.meta_data_ID,run_time from model_metrics mm, meta_data md
 where  mm.meta_data_ID=md.meta_data_ID
group by meta_data_ID;
```

count(model_ID)	meta_data_ID	run_time
18	1	500
25	2	1000
34	3	1500
47	4	2000
51	5	2500

```sql
--  Which run has the best model
select model_ID, run_time from hyperparameter_db14.model_metrics mm, hyperparameter_db14.meta_data md
where mm.meta_data_ID = md.meta_data_ID
 and auc=(select max(auc) from model_metrics);
```

model_ID	run_time
1	2500

```
-- 8.  For run time 1500 which model id has the ideal auc value
select model_ID, run_time from model_metrics  mm, meta_data md
where mm.meta_data_ID=md.meta_data_ID and auc=(select max(auc) from model_metrics);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

model_ID	run_time
1	2500

Functions

A function is a special kind stored program that returns a single value. Functions are used to encapsulate common formulas or business rules that are reusable among SQL statements or stored programs. It helps improve the readability and maintainability of the procedural code.

The following illustrates the syntax for creating a new function:

```
CREATE FUNCTION function_name(param1,param2,…)
    RETURNS datatype
   [NOT] DETERMINISTIC
 Statements
```

Below are the functions for the database:

```
-- 3.  Return run_id for highest run time
DELIMITER $$
CREATE FUNCTION FUN1()
RETURNS TEXT
DETERMINISTIC
BEGIN
DECLARE func TEXT;
select run_id INTO func from hyperparameter_db14.meta_data where run_time=
(select max(run_time) from hyperparameter_db14.meta_data);
RETURN func;
END $$
DELIMITER ;
select FUN1();
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

FUN1()
3oTEV9hn4

```sql
-- 4.  Return dataset_id for highest run time
DELIMITER $$
CREATE FUNCTION FUN2()
RETURNS INTEGER
DETERMINISTIC
BEGIN
DECLARE func integer;
 select dataset_id into func from hyperparameter_db14.meta_data where run_time= (select max(run_time)
 from hyperparameter_db14.meta_data);
RETURN func;
END $$
DELIMITER ;
select FUN2();
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

FUN2()
1

```sql
-- 5.  Highest logloss value for model id 5 for run time 2500
DELIMITER $$
CREATE FUNCTION FUN3()
RETURNS float
DETERMINISTIC
BEGIN
DECLARE func float;
select max(logloss) INTO func from hyperparameter_db14.model_metrics mm, hyperparameter_db14.meta_data md
where mM.meta_data_ID=md.meta_data_ID and model_ID=5 and run_time=2500;
RETURN func;
END $$
DELIMITER ;
select FUN3();
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

FUN3()
0.06681600958108902

```sql
-- 6.  Number of models for run time 2000
DELIMITER $$
CREATE FUNCTION FUN4()
RETURNS INTEGER
DETERMINISTIC
BEGIN
DECLARE func integer;
select count(model_ID) INTO func from hyperparameter_db14.model_metrics  mm, hyperparameter_db14.meta_data md
where mm.meta_data_ID=md.meta_data_ID and run_time=2000;
RETURN func;
END $$
DELIMITER ;
select FUN4();
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

FUN4()
47

Procedures

A stored procedure is a segment of declarative SQL statements stored inside the database catalog. A stored procedure can be invoked by triggers, other stored procedures, and applications such as Java, Python, PHP. Stored procedures increase the performance of the applications.

The following illustrates the syntax for creating a new function:

CREATE
  [DEFINER = user]
  PROCEDURE sp_name ([proc_parameter[,...]])
  [characteristic ...] routine_body

Below are the procedures for the database:

```
-- Average RMSE value for run time 1000
DELIMITER //
CREATE PROCEDURE PROC21()
BEGIN
select avg(RMSE),run_time,mm.model_ID from hyperparameter_db14.model_metrics mm , hyperparameter_db14.meta_data md
where mm.meta_data_ID = md.meta_data_ID
and run_time=1000;
END //
DELIMITER ;

CALL PROC21();
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

avg(RMSE)	run_time	model_ID
0.13124190320000004	1000	1

```
-- 12.  Display all the metrics values for run time 2500
 DELIMITER //
CREATE PROCEDURE PROC2()
BEGIN
select run_time,auc,RMSE,MSE,logloss,mean_per_class_error from hyperparameter_db14.model_metrics mm, hyperparameter_db14.meta_data md
where mm.meta_data_ID = md.meta_data_ID
and run_time=1000;
END //
DELIMITER ;
CALL PROC2();
drop procedure PROC2;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

run_time	auc	RMSE	MSE	logloss	mean_per_class_error
1000	0.825466329	0.118876642	0.014131656	0.065743315	0.335053974
1000	0.825077774	0.118005574	0.013925316	0.066296978	0.372464677
1000	0.823365718	0.121355077	0.014727055	0.068297189	0.366185974
1000	0.822387187	0.117911168	0.013903044	0.065602266	0.380755616
1000	0.819916268	0.119251903	0.014221016	0.066527222	0.341879035

Result 5

```sql
-- 13.  What are the start and end time of all runs
DELIMITER //
CREATE PROCEDURE PROC3()
BEGIN
select start_time, end_time, run_time from hyperparameter_db14.meta_data;
END //
DELIMITER ;
CALL PROC3();
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

start_time	end_time	run_time
1555918564	1555918564	500
1555920195	1555920195	1000
1555968866	1555968866	1500
1555972904	1555972904	2000
1555975812	1555975812	2500

```sql
-- Display MSE value for model 10 of run time 1000
DELIMITER //
CREATE PROCEDURE PROC24()
BEGIN
select MSE, mm.model_ID, run_time from hyperparameter_db14.model_metrics  mm , hyperparameter_db14.meta_data md
where mm.meta_data_ID = md.meta_data_ID
and run_time=1000;
END //
DELIMITER ;
CALL PROC24();
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

MSE	model_ID	run_time
0.014131656	1	1000
0.013925316	2	1000
0.014727055	3	1000

```sql
15. Highest MSE value for run time 500
DELIMITER //
CREATE PROCEDURE PROC25()
BEGIN
select max(MSE), run_time,mm.model_ID from hyperparameter_db14.model_metrics  mm ,
hyperparameter_db14.meta_data md
where mm.meta_data_ID = md.meta_data_ID and run_time=1000;
END //
DELIMITER ;
CALL PROC25();
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

max(MSE)	run_time	model_ID
0.05530114	1000	1

What are the run id of model id that has least mean_per_class_error

```sql
DELIMITER //
CREATE PROCEDURE PROC6()
BEGIN
 select run_id from hyperparameter_db14.meta_data md, hyperparameter_db14.model_metrics  mm
 where mm.meta_data_ID = md.meta_data_ID AND  mean_per_class_error= (select min(mean_per_class_error)
 from hyperparameter_db14.model_metrics);
END //
DELIMITER ;
CALL PROC6();
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

run_id
PVxu8FKke

```sql
    -- Return run_id for lowest run time
DELIMITER //
CREATE PROCEDURE PROC27()
BEGIN
select run_id, run_time lowest_run_time from hyperparameter_db14.model_metrics  mm ,
 hyperparameter_db14.meta_data md
where mm.meta_data_ID = md.meta_data_ID
and run_time= (select min(run_time) from hyperparameter_db14.meta_data);
END //
DELIMITER ;
CALL PROC27();
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

run_id	lowest_run_time
wtiOmXHNa	500

```sql
    Return dataset_id for lowest run time
DELIMITER //
CREATE PROCEDURE PROC28()
BEGIN
select dataset_id, run_time lowest_run_time from hyperparameter_db14.meta_data
 where run_time= (select min(run_time) from hyperparameter_db14.meta_data);
END //
DELIMITER ;
CALL PROC28();
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

dataset_id	lowest_run_time
1	500

```
-- 19.  Display the parameters for travel insurance dataset.
DELIMITER //
CREATE PROCEDURE PROC9()
BEGIN
select dataset_ID, dataset dataset_Name, dataset_type,total_columns, total_records from dataset;
END //
DELIMITER ;
CALL PROC9();
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

dataset_ID	dataset_Name	dataset_type	total_columns	total_records
1	TRAVEL INSURANCE	csv	11	63326

For run time 2500 which model id has idea auc value

```

DELIMITER //
CREATE PROCEDURE PROC10()
BEGIN
Select model_ID, run_time from model_metrics mm, meta_data md
Where mm.meta_data_ID=md.meta_data_ID and auc = (select max(auc) from model_metrics);
END //
DELIMITER ;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

model_ID	run_time
1	2500

Views

Views are stored queries that when invoked produce a result set. A view acts as a virtual table.and Views are used to restrict a data

The following illustrates the syntax for creating a new function:

```
CREATE
  [ALGORITHM = {MERGE  | TEMPTABLE | UNDEFINED}]
VIEW view_name [(column_list)]
AS
select-statement;
```

Below are the views for the database:

```sql
-- 1.   Display run time, stat time and end time for travel insurance dataset
create view v1 as select run_time, start_time, end_time from hyperparameter_db14.meta_data;
select * from v1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

run_time	start_time	end_time
500	1555918564	1555918564
1000	1555920195	1555920195
1500	1555968866	1555968866
2000	1555972904	1555972904
2500	1555975812	1555975812

```sql
-- 2.   Display all the models for run time 1500
create view v2 as select model_ID from hyperparameter_db14.model_metrics mm, hyperparameter_db14.meta_data md
where mm.meta_data_ID=md.meta_data_ID
and run_time=1500;

select * from v2;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

model_ID
1
2
3
4
5

```sql
-- 3.   Create a view having run time and its models where RMSE value is greater than 0.5
create view v3 as select run_time, model_ID from hyperparameter_db14.model_metrics mm,
 hyperparameter_db14.meta_data md where mm.meta_data_ID=md.meta_data_ID
and RMSE > 0.05;

select * from v3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

run_time	model_ID
500	1
500	2
500	3
500	4
500	5

```sql
-- 5.  Display all the metrics values for run time 500
create view v4 as select auc,logloss,mean_per_class_error,rmse,mse from hyperparameter_db14.model_metrics m ,
 hyperparameter_db14.meta_data md where m.meta_data_ID=md.meta_data_ID
and run_time=500;

select * from v4;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

auc	logloss	mean_per_class_error	rmse	mse
0.830198732	0.067567692	0.368385551	0.121168588	0.014681827
0.824804055	0.065958618	0.364623942	0.119122526	0.014190176
0.819261523	0.066604943	0.395916695	0.119456081	0.014269755
0.812347204	0.067957951	0.347793663	0.119143168	0.014195094
0.810830103	0.066057545	0.389438204	0.118158943	0.013961536

```sql
-- 8.  Display the run time and models with its auc valuese
create view v5 as select run_time, model_ID, auc from hyperparameter_db14.model_metrics mm,
 hyperparameter_db14.meta_data md
where mm.meta_data_ID=md.meta_data_ID;

select * from v5;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

run_time	model_ID	auc
500	1	0.830198732
500	2	0.824804055
500	3	0.819261523
500	4	0.812347204
500	5	0.810830103

Indexes

Below are the indexes for the database:

1 Create index index1_ on dataset (dataset_ID);

```sql
DELIMITER $$
CREATE FUNCTION FUN6()
RETURNS INTEGER
DETERMINISTIC
BEGIN
DECLARE func integer;
select count(model_ID) INTO func from hyperparameter_db14.model_metrics  mm, hyperparameter_db14.meta_data md
where mm.meta_data_ID=md.meta_data_ID and run_time=2000;
RETURN func;
END $$
DELIMITER ;
```

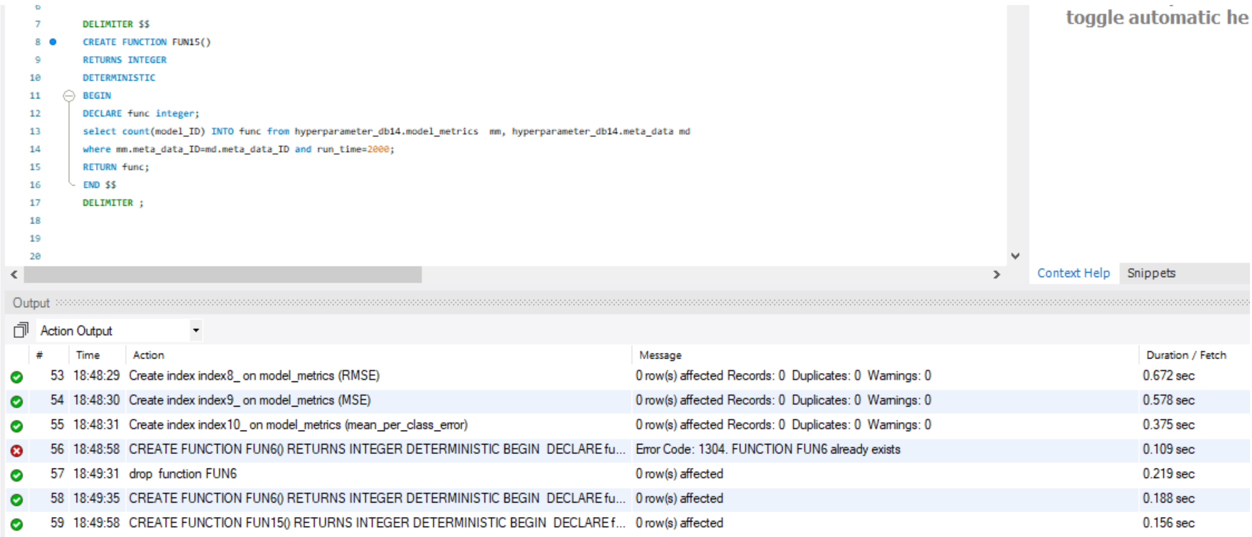
Context Help | Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
11	18:08:14	select max(auc) from model_metrics LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
12	18:08:33	select model_ID from model_metrics mm, meta_data md where mm.meta_data_ID=md.meta_...	0 row(s) returned	0.000 sec / 0.000 sec
13	18:08:54	select model_ID, run_time from model_metrics mm, meta_data md where mm.meta_data_ID=...	1 row(s) returned	0.000 sec / 0.000 sec
14	18:11:41	SELECT * FROM hyperparameter_db14.dataset LIMIT 0, 1000	1 row(s) returned	0.031 sec / 0.000 sec
15	18:11:57	CREATE FUNCTION FUN6() RETURNS INTEGER DETERMINISTIC BEGIN DECLARE fu...	0 row(s) affected	0.453 sec

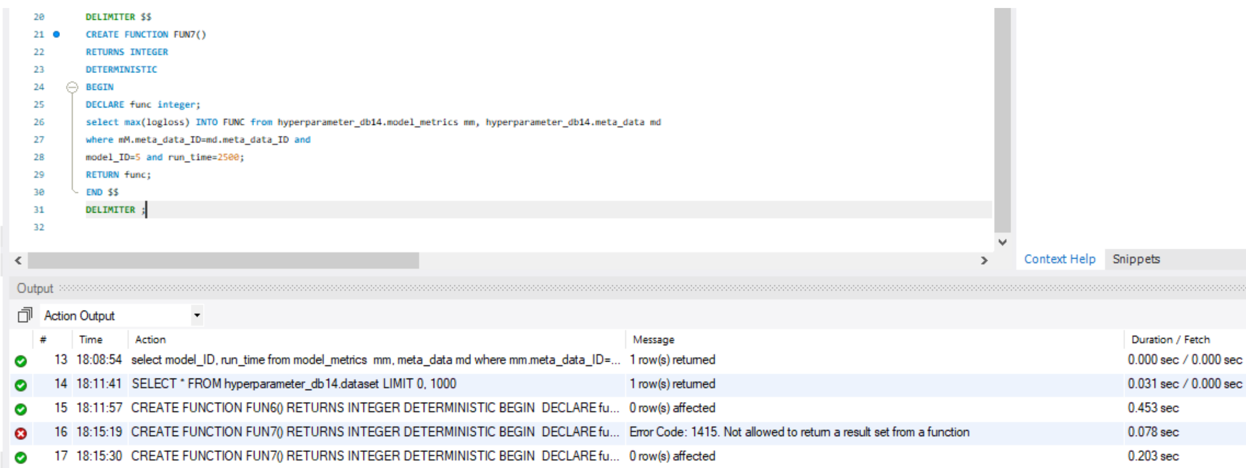
AFTER



```
DELIMITER $$
CREATE FUNCTION FUN15()
RETURNS INTEGER
DETERMINISTIC
BEGIN
DECLARE func integer;
select count(model_ID) INTO func from hyperparameter_db14.model_metrics  mm, hyperparameter_db14.meta_data md
where mm.meta_data_ID=md.meta_data_ID and run_time=2000;
RETURN func;
END $$
DELIMITER ;
```


#	Time	Action	Message	Duration / Fetch
53	18:48:29	Create index index8_ on model_metrics (RMSE)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.672 sec
54	18:48:30	Create index index9_ on model_metrics (MSE)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.578 sec
55	18:48:31	Create index index10_ on model_metrics (mean_per_class_error)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.375 sec
56	18:48:58	CREATE FUNCTION FUN6() RETURNS INTEGER DETERMINISTIC BEGIN DECLARE fu...	Error Code: 1304. FUNCTION FUN6 already exists	0.109 sec
57	18:49:31	drop function FUN6	0 row(s) affected	0.219 sec
58	18:49:35	CREATE FUNCTION FUN6() RETURNS INTEGER DETERMINISTIC BEGIN DECLARE fu...	0 row(s) affected	0.188 sec
59	18:49:58	CREATE FUNCTION FUN15() RETURNS INTEGER DETERMINISTIC BEGIN DECLARE f...	0 row(s) affected	0.156 sec

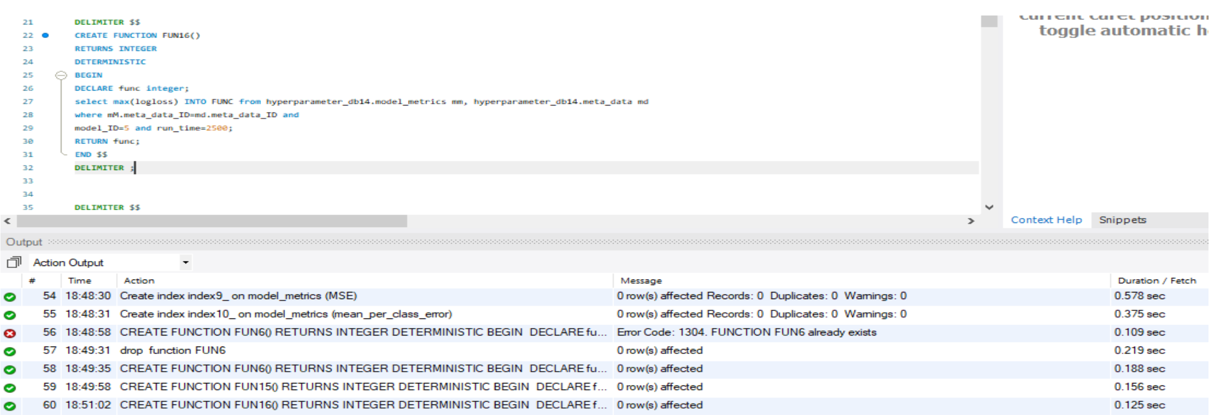
2 Create index index2_ on meta_data (meta_data_ID);



```
DELIMITER $$
CREATE FUNCTION FUN7()
RETURNS INTEGER
DETERMINISTIC
BEGIN
DECLARE func integer;
select max(logloss) INTO FUNC from hyperparameter_db14.model_metrics mm, hyperparameter_db14.meta_data md
where mM.meta_data_ID=md.meta_data_ID and
model_ID=5 and run_time=2500;
RETURN func;
END $$
DELIMITER ;
```


#	Time	Action	Message	Duration / Fetch
13	18:08:54	select model_ID, run_time from model_metrics mm, meta_data md where mm.meta_data_ID=...	1 row(s) returned	0.000 sec / 0.000 sec
14	18:11:41	SELECT * FROM hyperparameter_db14.dataset LIMIT 0, 1000	1 row(s) returned	0.031 sec / 0.000 sec
15	18:11:57	CREATE FUNCTION FUN6() RETURNS INTEGER DETERMINISTIC BEGIN DECLARE fu...	0 row(s) affected	0.453 sec
16	18:15:19	CREATE FUNCTION FUN7() RETURNS INTEGER DETERMINISTIC BEGIN DECLARE fu...	Error Code: 1415. Not allowed to return a result set from a function	0.078 sec
17	18:15:30	CREATE FUNCTION FUN7() RETURNS INTEGER DETERMINISTIC BEGIN DECLARE fu...	0 row(s) affected	0.203 sec

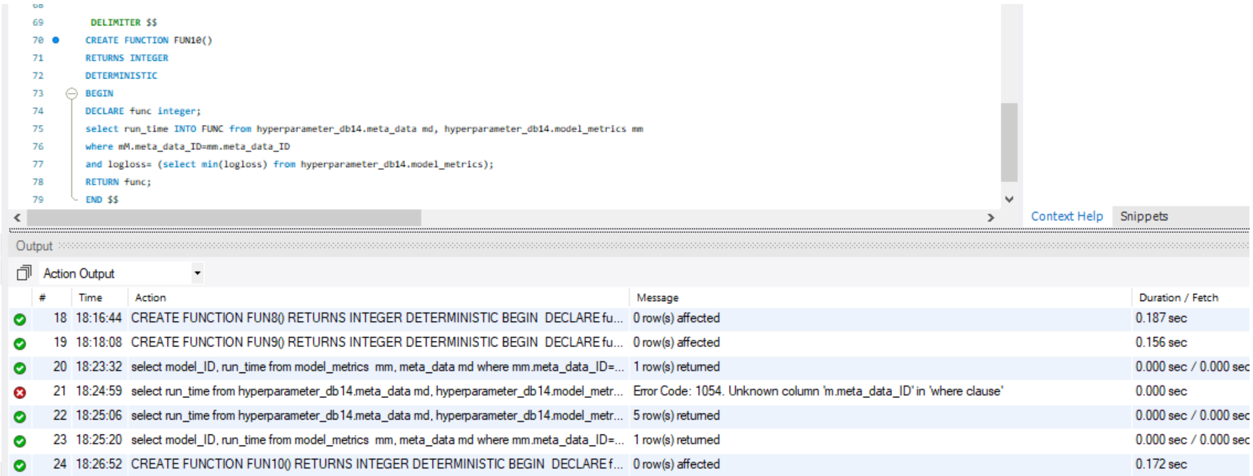
AFTER



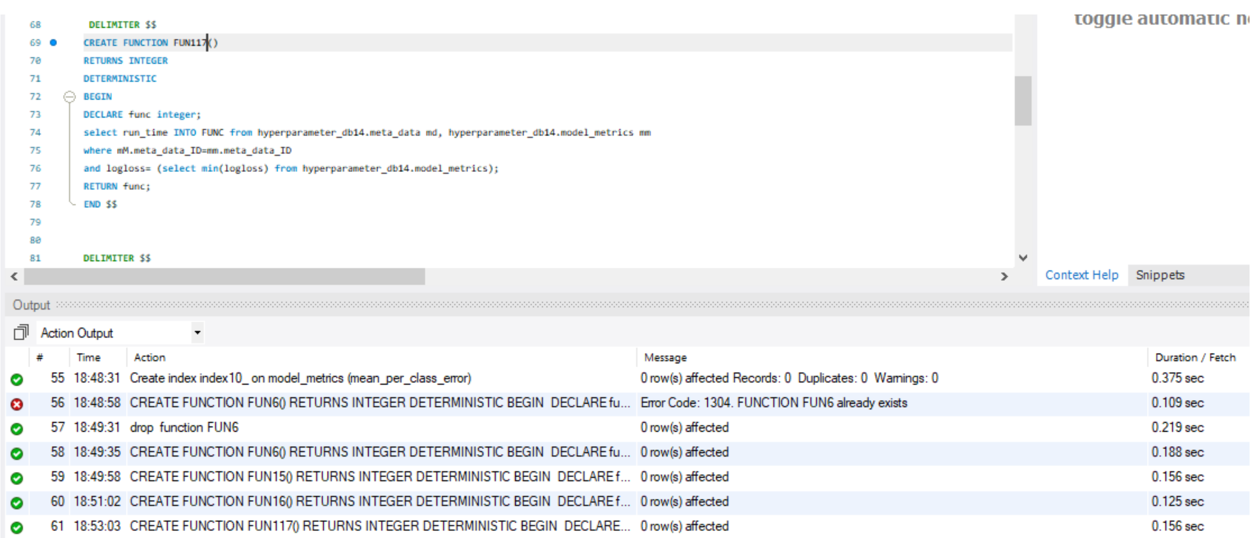
```
DELIMITER $$
CREATE FUNCTION FUN16()
RETURNS INTEGER
DETERMINISTIC
BEGIN
DECLARE func integer;
select max(logloss) INTO FUNC from hyperparameter_db14.model_metrics mm, hyperparameter_db14.meta_data md
where mM.meta_data_ID=md.meta_data_ID and
model_ID=5 and run_time=2500;
RETURN func;
END $$
DELIMITER ;


DELIMITER $$
```


#	Time	Action	Message	Duration / Fetch
54	18:48:30	Create index index9_ on model_metrics (MSE)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.578 sec
55	18:48:31	Create index index10_ on model_metrics (mean_per_class_error)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.375 sec
56	18:48:58	CREATE FUNCTION FUN6() RETURNS INTEGER DETERMINISTIC BEGIN DECLARE fu...	Error Code: 1304. FUNCTION FUN6 already exists	0.109 sec
57	18:49:31	drop function FUN6	0 row(s) affected	0.219 sec
58	18:49:35	CREATE FUNCTION FUN6() RETURNS INTEGER DETERMINISTIC BEGIN DECLARE fu...	0 row(s) affected	0.188 sec
59	18:49:58	CREATE FUNCTION FUN15() RETURNS INTEGER DETERMINISTIC BEGIN DECLARE f...	0 row(s) affected	0.156 sec
60	18:51:02	CREATE FUNCTION FUN16() RETURNS INTEGER DETERMINISTIC BEGIN DECLARE f...	0 row(s) affected	0.125 sec

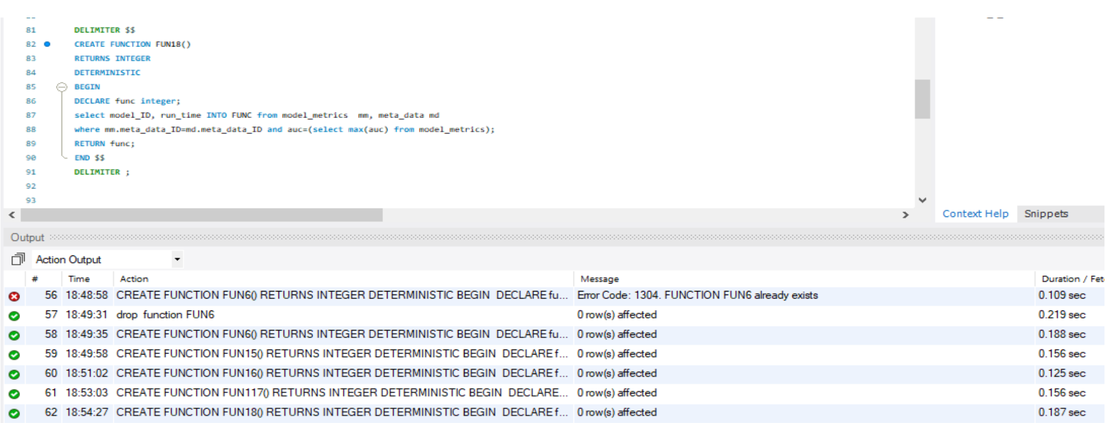
3 Create index index3_ on model_metrics (model_metrics_ID);



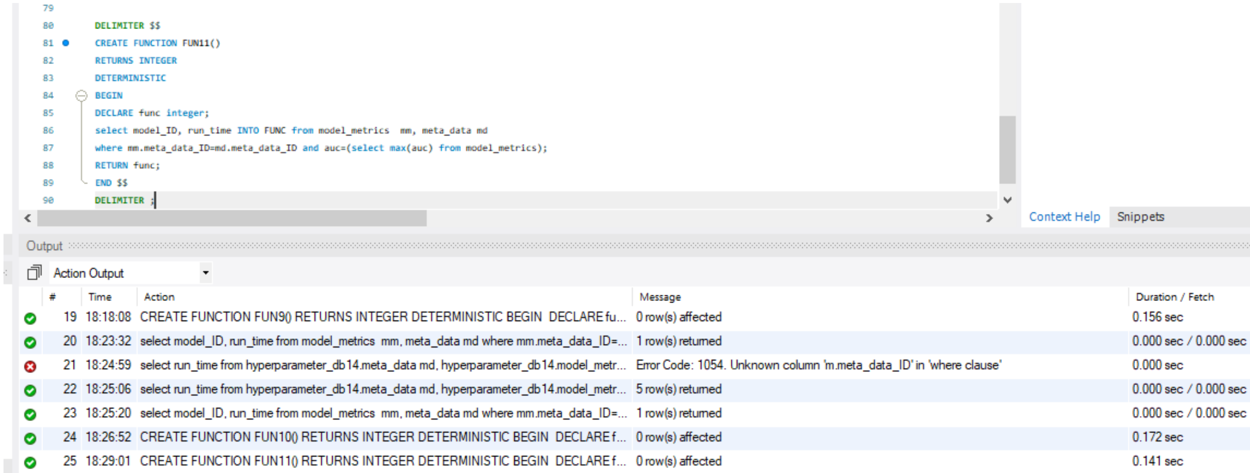
AFTER



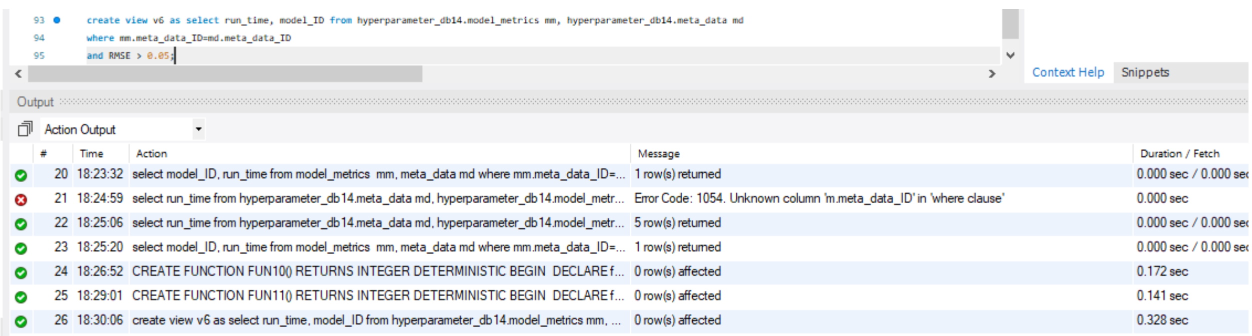
4 Create index index4_ on hyperparameter (hp_ID);



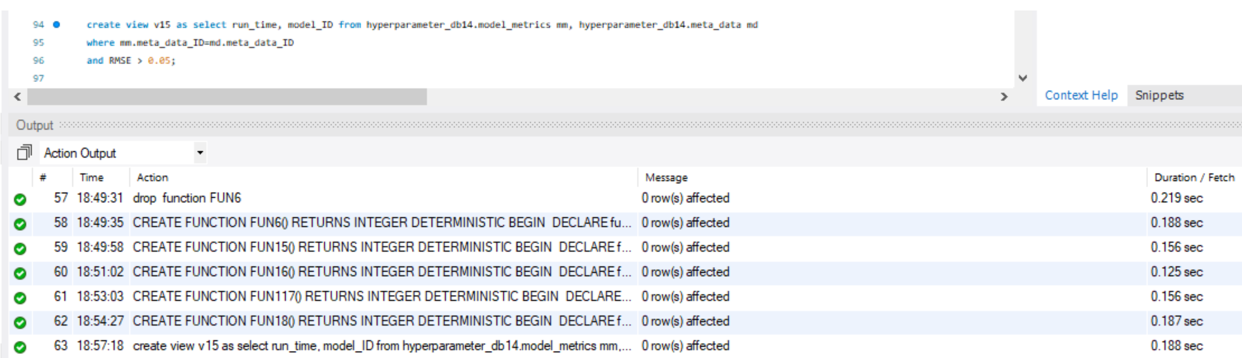
AFTER



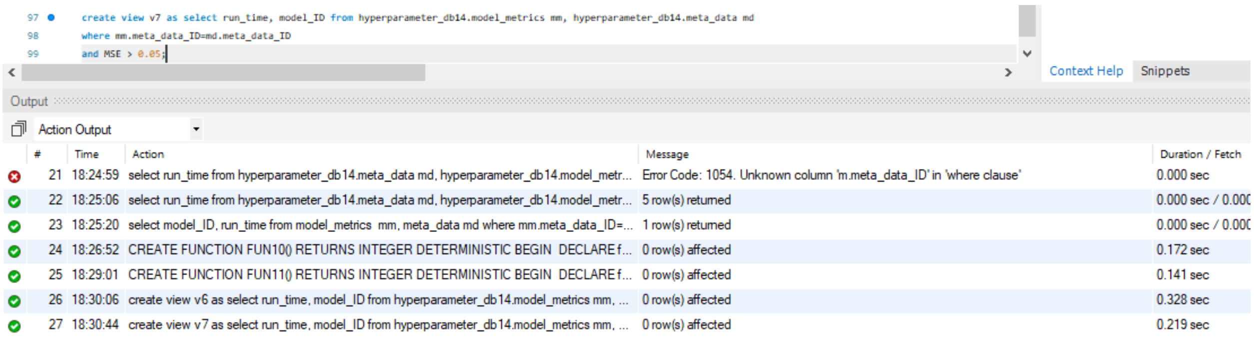
5 Create index index5_ on meta_data(run_time);



AFTER

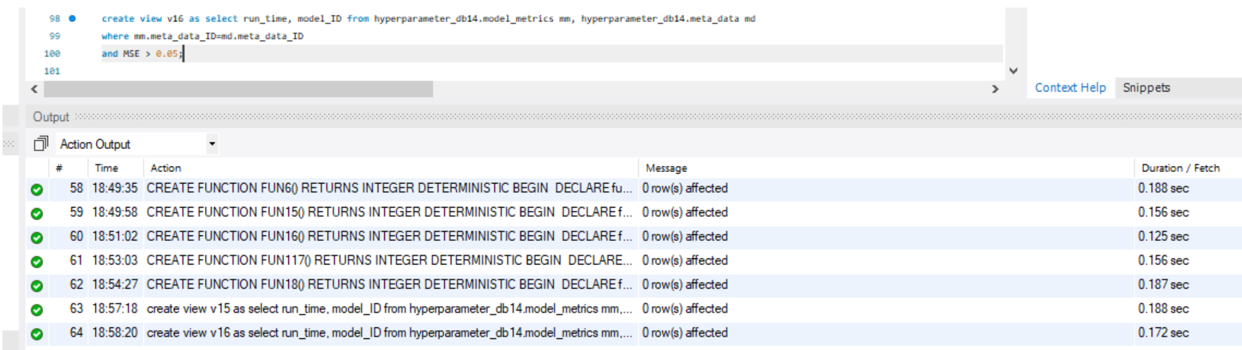


6 Create index index6_ on model_metrics (logloss);



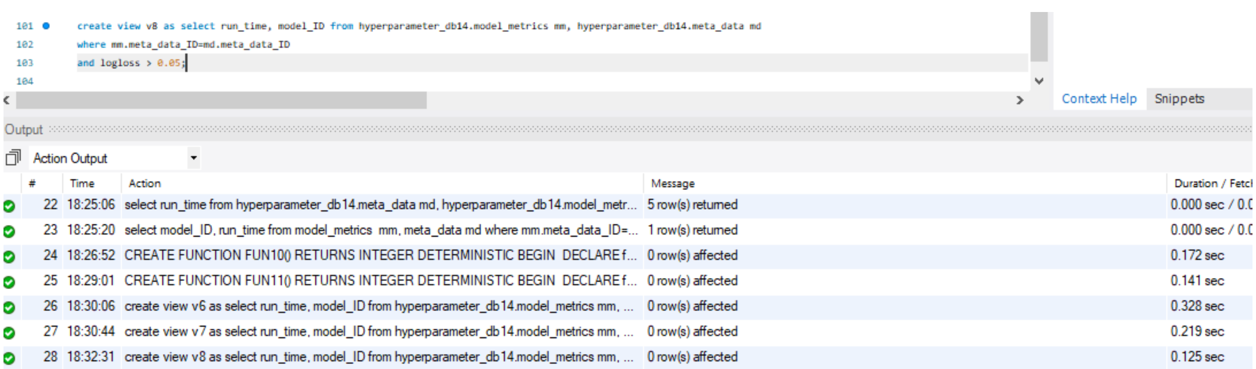
	#	Time	Action	Message	Duration / Fetch
✖	21	18:24:59	select run_time from hyperparameter_db14.meta_data md, hyperparameter_db14.model_metr...	Error Code: 1054. Unknown column 'm.meta_data_ID' in 'where clause'	0.000 sec
✔	22	18:25:06	select run_time from hyperparameter_db14.meta_data md, hyperparameter_db14.model_metr...	5 row(s) returned	0.000 sec / 0.000
✔	23	18:25:20	select model_ID, run_time from model_metrics mm, meta_data md where mm.meta_data_ID=...	1 row(s) returned	0.000 sec / 0.000
✔	24	18:26:52	CREATE FUNCTION FUN10() RETURNS INTEGER DETERMINISTIC BEGIN DECLARE f...	0 row(s) affected	0.172 sec
✔	25	18:29:01	CREATE FUNCTION FUN11() RETURNS INTEGER DETERMINISTIC BEGIN DECLARE f...	0 row(s) affected	0.141 sec
✔	26	18:30:06	create view v6 as select run_time, model_ID from hyperparameter_db14.model_metrics mm, ...	0 row(s) affected	0.328 sec
✔	27	18:30:44	create view v7 as select run_time, model_ID from hyperparameter_db14.model_metrics mm, ...	0 row(s) affected	0.219 sec

AFTER



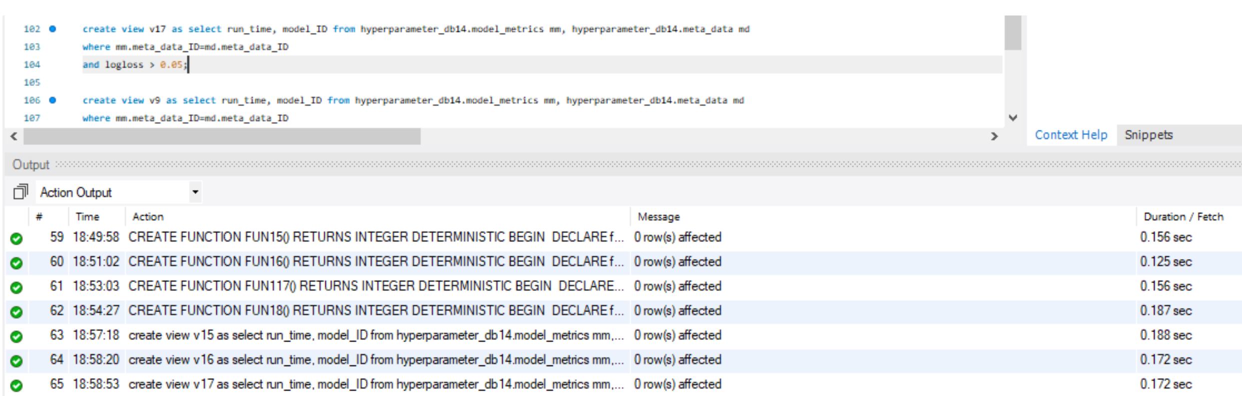
	#	Time	Action	Message	Duration / Fetch
✔	58	18:49:35	CREATE FUNCTION FUN6() RETURNS INTEGER DETERMINISTIC BEGIN DECLARE fu...	0 row(s) affected	0.188 sec
✔	59	18:49:58	CREATE FUNCTION FUN15() RETURNS INTEGER DETERMINISTIC BEGIN DECLARE f...	0 row(s) affected	0.156 sec
✔	60	18:51:02	CREATE FUNCTION FUN16() RETURNS INTEGER DETERMINISTIC BEGIN DECLARE f...	0 row(s) affected	0.125 sec
✔	61	18:53:03	CREATE FUNCTION FUN117() RETURNS INTEGER DETERMINISTIC BEGIN DECLARE...	0 row(s) affected	0.156 sec
✔	62	18:54:27	CREATE FUNCTION FUN18() RETURNS INTEGER DETERMINISTIC BEGIN DECLARE f...	0 row(s) affected	0.187 sec
✔	63	18:57:18	create view v15 as select run_time, model_ID from hyperparameter_db14.model_metrics mm,...	0 row(s) affected	0.188 sec
✔	64	18:58:20	create view v16 as select run_time, model_ID from hyperparameter_db14.model_metrics mm,...	0 row(s) affected	0.172 sec

7 Create index index7_ on model_metrics (auc);



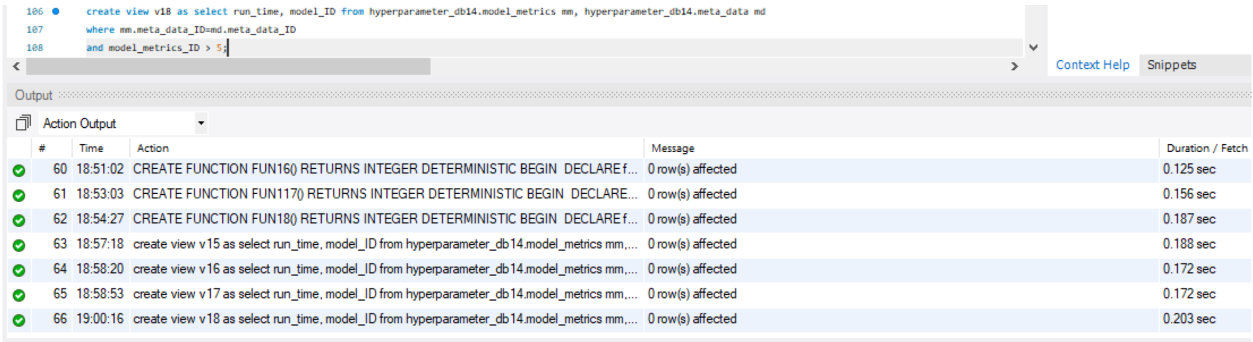
	#	Time	Action	Message	Duration / Fetch
✔	22	18:25:06	select run_time from hyperparameter_db14.meta_data md, hyperparameter_db14.model_metr...	5 row(s) returned	0.000 sec / 0.0
✔	23	18:25:20	select model_ID, run_time from model_metrics mm, meta_data md where mm.meta_data_ID=...	1 row(s) returned	0.000 sec / 0.0
✔	24	18:26:52	CREATE FUNCTION FUN10() RETURNS INTEGER DETERMINISTIC BEGIN DECLARE f...	0 row(s) affected	0.172 sec
✔	25	18:29:01	CREATE FUNCTION FUN11() RETURNS INTEGER DETERMINISTIC BEGIN DECLARE f...	0 row(s) affected	0.141 sec
✔	26	18:30:06	create view v6 as select run_time, model_ID from hyperparameter_db14.model_metrics mm, ...	0 row(s) affected	0.328 sec
✔	27	18:30:44	create view v7 as select run_time, model_ID from hyperparameter_db14.model_metrics mm, ...	0 row(s) affected	0.219 sec
✔	28	18:32:31	create view v8 as select run_time, model_ID from hyperparameter_db14.model_metrics mm, ...	0 row(s) affected	0.125 sec

AFTER



	#	Time	Action	Message	Duration / Fetch
✔	59	18:49:58	CREATE FUNCTION FUN15() RETURNS INTEGER DETERMINISTIC BEGIN DECLARE f...	0 row(s) affected	0.156 sec
✔	60	18:51:02	CREATE FUNCTION FUN16() RETURNS INTEGER DETERMINISTIC BEGIN DECLARE f...	0 row(s) affected	0.125 sec
✔	61	18:53:03	CREATE FUNCTION FUN117() RETURNS INTEGER DETERMINISTIC BEGIN DECLARE...	0 row(s) affected	0.156 sec
✔	62	18:54:27	CREATE FUNCTION FUN18() RETURNS INTEGER DETERMINISTIC BEGIN DECLARE f...	0 row(s) affected	0.187 sec
✔	63	18:57:18	create view v15 as select run_time, model_ID from hyperparameter_db14.model_metrics mm,...	0 row(s) affected	0.188 sec
✔	64	18:58:20	create view v16 as select run_time, model_ID from hyperparameter_db14.model_metrics mm,...	0 row(s) affected	0.172 sec
✔	65	18:58:53	create view v17 as select run_time, model_ID from hyperparameter_db14.model_metrics mm,...	0 row(s) affected	0.172 sec

8 Create index index8_ on model_metrics (RMSE);



```
create view v18 as select run_time, model_ID from hyperparameter_db14.model_metrics mm, hyperparameter_db14.meta_data md
where mm.meta_data_ID=md.meta_data_ID
and model_metrics_ID > 5;
```

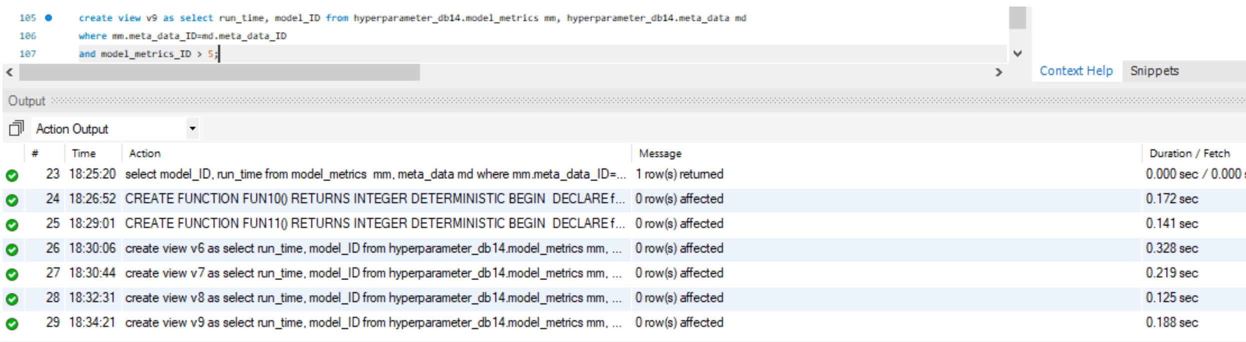
Context Help Snippets

Output

Action Output

	#	Time	Action	Message	Duration / Fetch
	60	18:51:02	CREATE FUNCTION FUN16() RETURNS INTEGER DETERMINISTIC BEGIN DECLARE f...	0 row(s) affected	0.125 sec
	61	18:53:03	CREATE FUNCTION FUN117() RETURNS INTEGER DETERMINISTIC BEGIN DECLARE...	0 row(s) affected	0.156 sec
	62	18:54:27	CREATE FUNCTION FUN18() RETURNS INTEGER DETERMINISTIC BEGIN DECLARE f...	0 row(s) affected	0.187 sec
	63	18:57:18	create view v15 as select run_time, model_ID from hyperparameter_db14.model_metrics mm,...	0 row(s) affected	0.188 sec
	64	18:58:20	create view v16 as select run_time, model_ID from hyperparameter_db14.model_metrics mm,...	0 row(s) affected	0.172 sec
	65	18:58:53	create view v17 as select run_time, model_ID from hyperparameter_db14.model_metrics mm,...	0 row(s) affected	0.172 sec
	66	19:00:16	create view v18 as select run_time, model_ID from hyperparameter_db14.model_metrics mm,...	0 row(s) affected	0.203 sec

AFTER



```
create view v9 as select run_time, model_ID from hyperparameter_db14.model_metrics mm, hyperparameter_db14.meta_data md
where mm.meta_data_ID=md.meta_data_ID
and model_metrics_ID > 5;
```

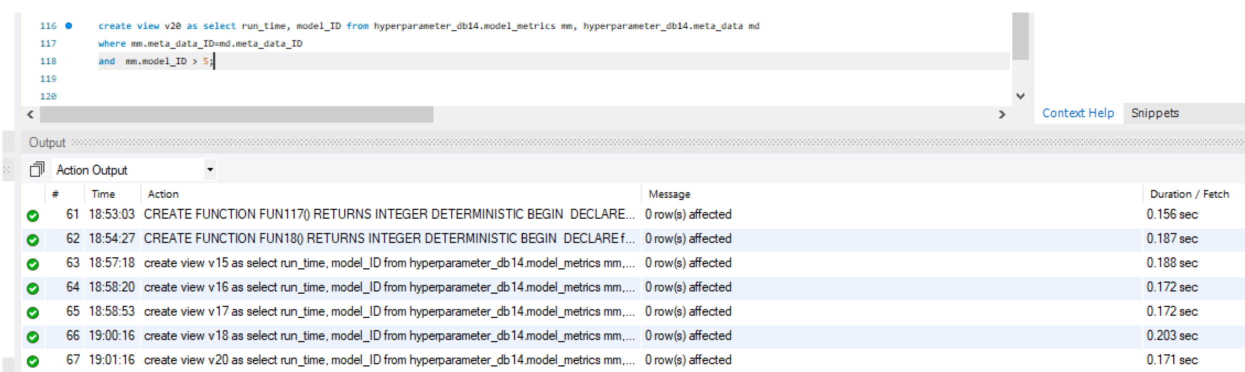
Context Help Snippets

Output

Action Output

	#	Time	Action	Message	Duration / Fetch
	23	18:25:20	select model_ID, run_time from model_metrics mm, meta_data md where mm.meta_data_ID=...	1 row(s) returned	0.000 sec / 0.000 s
	24	18:26:52	CREATE FUNCTION FUN10() RETURNS INTEGER DETERMINISTIC BEGIN DECLARE f...	0 row(s) affected	0.172 sec
	25	18:29:01	CREATE FUNCTION FUN11() RETURNS INTEGER DETERMINISTIC BEGIN DECLARE f...	0 row(s) affected	0.141 sec
	26	18:30:06	create view v6 as select run_time, model_ID from hyperparameter_db14.model_metrics mm, ...	0 row(s) affected	0.328 sec
	27	18:30:44	create view v7 as select run_time, model_ID from hyperparameter_db14.model_metrics mm, ...	0 row(s) affected	0.219 sec
	28	18:32:31	create view v8 as select run_time, model_ID from hyperparameter_db14.model_metrics mm, ...	0 row(s) affected	0.125 sec
	29	18:34:21	create view v9 as select run_time, model_ID from hyperparameter_db14.model_metrics mm, ...	0 row(s) affected	0.188 sec

9 Create index index9_ on model_metrics (MSE);



```
create view v20 as select run_time, model_ID from hyperparameter_db14.model_metrics mm, hyperparameter_db14.meta_data md
where mm.meta_data_ID=md.meta_data_ID
and  mm.model_ID > 5;
```

Context Help Snippets

Output

Action Output

	#	Time	Action	Message	Duration / Fetch
	61	18:53:03	CREATE FUNCTION FUN117() RETURNS INTEGER DETERMINISTIC BEGIN DECLARE...	0 row(s) affected	0.156 sec
	62	18:54:27	CREATE FUNCTION FUN18() RETURNS INTEGER DETERMINISTIC BEGIN DECLARE f...	0 row(s) affected	0.187 sec
	63	18:57:18	create view v15 as select run_time, model_ID from hyperparameter_db14.model_metrics mm,...	0 row(s) affected	0.188 sec
	64	18:58:20	create view v16 as select run_time, model_ID from hyperparameter_db14.model_metrics mm,...	0 row(s) affected	0.172 sec
	65	18:58:53	create view v17 as select run_time, model_ID from hyperparameter_db14.model_metrics mm,...	0 row(s) affected	0.172 sec
	66	19:00:16	create view v18 as select run_time, model_ID from hyperparameter_db14.model_metrics mm,...	0 row(s) affected	0.203 sec
	67	19:01:16	create view v20 as select run_time, model_ID from hyperparameter_db14.model_metrics mm,...	0 row(s) affected	0.171 sec

AFTER

```sql
create view v11 as select run_time, model_ID from hyperparameter_db14.model_metrics mm, hyperparameter_db14.meta_data md
where mm.meta_data_ID=md.meta_data_ID
and  mm.model_ID > 5;
```

Context Help | Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
26	18:30:06	create view v6 as select run_time, model_ID from hyperparameter_db14.model_metrics mm, ...	0 row(s) affected	0.328 sec
27	18:30:44	create view v7 as select run_time, model_ID from hyperparameter_db14.model_metrics mm, ...	0 row(s) affected	0.219 sec
28	18:32:31	create view v8 as select run_time, model_ID from hyperparameter_db14.model_metrics mm, ...	0 row(s) affected	0.125 sec
29	18:34:21	create view v9 as select run_time, model_ID from hyperparameter_db14.model_metrics mm, ...	0 row(s) affected	0.188 sec
30	18:36:12	create view v10 as select run_time, model_ID from hyperparameter_db14.model_metrics mm,...	Error Code: 1052. Column 'meta_data_ID' in where clause is ambiguous	0.000 sec
31	18:36:24	create view v10 as select run_time, model_ID from hyperparameter_db14.model_metrics mm,...	0 row(s) affected	0.157 sec
32	18:37:24	create view v11 as select run_time, model_ID from hyperparameter_db14.model_metrics mm,...	0 row(s) affected	0.141 sec

10 Create index index10_ on model_metrics (mean_per_class_error);

```sql
create view v12 as select auc,logloss from  hyperparameter_db14.hyperparameter h,hyperparameter_db14.model_metrics mm
where mm.model_metrics_ID=h.model_metrics_ID
and  hp_ID > 5;
```

Context Help | Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
29	18:34:21	create view v9 as select run_time, model_ID from hyperparameter_db14.model_metrics mm, ...	0 row(s) affected	0.188 sec
30	18:36:12	create view v10 as select run_time, model_ID from hyperparameter_db14.model_metrics mm,...	Error Code: 1052. Column 'meta_data_ID' in where clause is ambiguous	0.000 sec
31	18:36:24	create view v10 as select run_time, model_ID from hyperparameter_db14.model_metrics mm,...	0 row(s) affected	0.157 sec
32	18:37:24	create view v11 as select run_time, model_ID from hyperparameter_db14.model_metrics mm,...	0 row(s) affected	0.141 sec
33	18:39:09	create view v12 as select * from hyperparameter_db14.model_metrics mm, hyperparameter_d...	Error Code: 1060. Duplicate column name 'model_metrics_ID'	0.000 sec
34	18:40:00	create view v12 as select * from hyperparameter_db14.hyperparameter h,hyperparameter_d...	Error Code: 1060. Duplicate column name 'model_metrics_ID'	0.000 sec
35	18:40:43	create view v12 as select auc,logloss from hyperparameter_db14.hyperparameter h,hyperpa...	0 row(s) affected	0.266 sec

AFTER

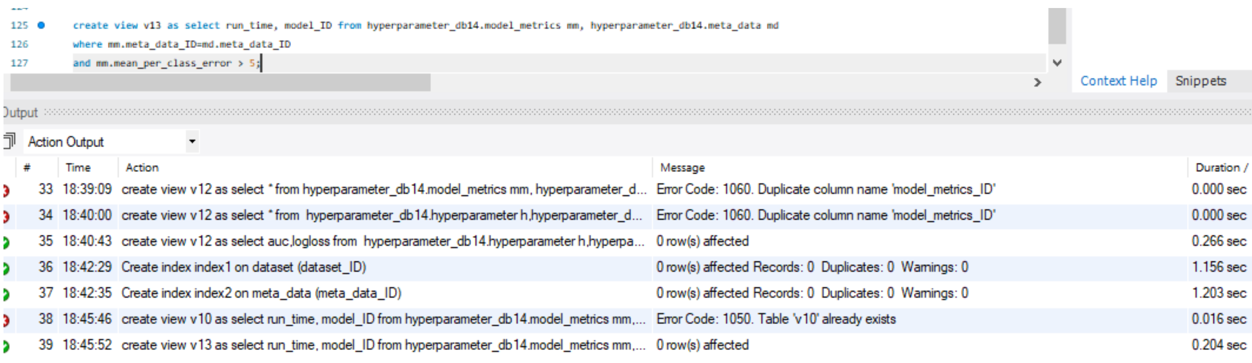
```sql
create view v21 as select auc,logloss from  hyperparameter_db14.hyperparameter h,hyperparameter_db14.model_metrics mm
where mm.model_metrics_ID=h.model_metrics_ID
and  hp_ID > 5;
```

Context Help | Snippets

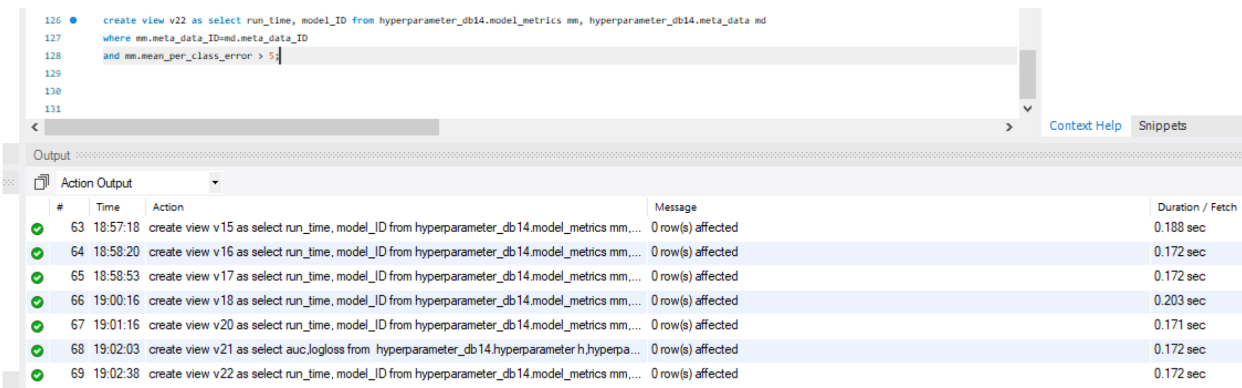
Output

Action Output

#	Time	Action	Message	Duration / Fetch
62	18:54:27	CREATE FUNCTION FUN18() RETURNS INTEGER DETERMINISTIC BEGIN DECLARE f...	0 row(s) affected	0.187 sec
63	18:57:18	create view v15 as select run_time, model_ID from hyperparameter_db14.model_metrics mm,...	0 row(s) affected	0.188 sec
64	18:58:20	create view v16 as select run_time, model_ID from hyperparameter_db14.model_metrics mm,...	0 row(s) affected	0.172 sec
65	18:58:53	create view v17 as select run_time, model_ID from hyperparameter_db14.model_metrics mm,...	0 row(s) affected	0.172 sec
66	19:00:16	create view v18 as select run_time, model_ID from hyperparameter_db14.model_metrics mm,...	0 row(s) affected	0.203 sec
67	19:01:16	create view v20 as select run_time, model_ID from hyperparameter_db14.model_metrics mm,...	0 row(s) affected	0.171 sec
68	19:02:03	create view v21 as select auc,logloss from hyperparameter_db14.hyperparameter h,hyperpa...	0 row(s) affected	0.172 sec



AFTER



Conclusion

Hyperparameter database is built by running several hyperparameter values on Travel Insurance dataset. Data fetched from the dataset is structured into different tables analyzing the relationships between the values.

Using H20 software, for five run times (500, 1000, 1500, 2000, 2500), respective models and its metrics values are generated. Metrics indicate the absolute fit of the model to actual data. Smaller the metrics values, closer we are to finding the line of best fit. Thus, models generated their own set of hyperparameters. Hence, model architecture is defined for storing the hyperparameter values in the database.

References

- https://en.wikipedia.org/wiki/Hyperparameter_(machine_learning)
- https://towardsdatascience.com/top-sources-for-machine-learning-datasets-bb6d0dc3378b
- https://en.wikipedia.org/wiki/List_of_datasets_for_machine-learning_research

- https://medium.com/@alexandraj777/top-5-mistakes-data-scientists-make-with-hyperparameter-optimization-and-how-to-prevent-them-767638b245f8
- https://stats.stackexchange.com/questions/297337/what-are-some-of-the-disavantage-of-bayesian-hyper-parameter-optimization
- https://github.com/skunkworksneu/Projects
- http://uksanjay.blogspot.com/2012/06/difference-between-conceptual-logical.html
- https://www.smartdraw.com/entity-relationship-diagram/
- http://www.mysqltutorial.org/create-sql-views-mysql.aspx
- http://www.mysqltutorial.org/mysql-stored-function/
- http://www.mysqltutorial.org/introduction-to-sql-stored-procedures.aspx
- https://dev.mysql.com/doc/refman/8.0/en/create-procedure.html

License